Question-4

**Implementation:**

**Graph Representation**: The graph is represented using an adjacency matrix (adjacent), where adjacent[i][j] = 1 indicates a road between locations i and j.

**Pathfinding Algorithms**: Two algorithms are implemented for pathfinding:

* 1. mininteractionpath: Uses BFS to find the path with the minimum number of intersections between a source and a destination.
  2. nonoverlappath: Finds a non-overlapping path by ensuring that previously visited nodes and edges are blocked.

**Dynamic Memory Allocation**: Various arrays like parent, visited, blockednode, blockededge are dynamically allocated and cleaned up to handle varying graph sizes.

Data Structure Used:

**Adjacency Matrix**: The graph is stored as an adjacency matrix (adjacent) of size location x location.

**Queue (for BFS)**: BFS is implemented using a queue to explore nodes in a level-wise manner.

**Parent Array**: Used to trace the path back from the destination to the source.

**Visited Array**: Ensures each node is visited only once.

Use of data structures:

**Graph (adjacent matrix)**: Used for graph representation. Each edge is checked in O(1) time due to the matrix structure.

**Queue**: Used in both pathfinding algorithms to explore nodes in a breadth-first manner.

**Parent Array**: Keeps track of the previous node for each visited node, enabling the reconstruction of the path.

**Blocked Node/Edge Arrays**: Used to track nodes and edges that are disallowed during the second path search.

Conclusion:

We have used bfs for traversal and find the minimum path from source to destination why we have chosed bfs is it performance better than dfs when it comes to finding minimum path .Dijkstra is more complex it is uses priority queue which is more complex. If a path is found, it traces back from the destination to the source and prints the path.

After finding the minimum interaction path, this function finds a second path between the same source and destination, avoiding nodes and edges used in the first path. It again uses BFS but checks for blocked nodes and edges.

Memory is dynamically allocated for arrays like the adjacency matrix, parent, and visited arrays, and all allocated memory is properly deallocated at the end of the program to prevent memory leaks.